Diversity and Locality in Multi-Component, Multi-Layer Predictive Systems: A Mutual Information Based Approach

Bassma Al-Jubouri(✉) and Bogdan Gabrys

Data Science Institute, Bournemouth University, Dorset BH12 5BB, UK
{baljubouri,bgabrys}@bournemouth.ac.uk
http://staffprofiles.bournemouth.ac.uk/display/i7217997
http://bogdan-gabrys.com

Abstract. This paper discusses the effect of locality and diversity among the base models of a Multi-Components Multi-Layer Predictive System (MCMLPS). A new ensemble method is introduced, where in the proposed architecture, the data instances are assigned to local regions using a conditional mutual information based on the similarity of their features. Furthermore, the outputs of the base models are weighted by this similarity metric. The proposed architecture has been tested on a number of data sets and its performance was compared to four benchmark algorithms. Moreover, the effect of changing three parameters of the proposed architecture has been tested and compared.

Keywords: Ensemble diversity · Ensemble methods · Local learning · Conditional mutual information · Feature selection

1 Introduction

Ensemble learning have shown many theoretical and practical benefits compared to the use of a single best model [13,18]. As opposed to using a single predictor, ensemble methods have statistical benefits acquired from combining the output of several predictors. It provides a divide and conquer strategy that a single predictor is incapable of achieving when the problem is too difficult and provides a more accurate representation of the data when the data is generated from different sources (data fusion).

An early example of the use of ensemble methods in literature is presented in [7], where the feature space is partitioned using two or more classifiers. In the nineties, two of the most widely used ensemble methods where proposed, these are: Boosting [19] and Bagging [3]. Schapire introduced Boosting algorithm in [19] where the author showed that a strong learner can be built by combining a number of weak learners. The introduction of Boosting has led to the development of AdaBoost and its many variations to solve multi-class and regression problems. Meanwhile, Breiman introduced Bagging in [3], where the base predictors are trained on bootstrap replicas of the training data. In addition to these

G. Cong et al. (Eds.): ADMA 2017, LNAI 10604, pp. 313–325, 2017.
https://doi.org/10.1007/978-3-319-69179-4_22

two algorithms, many well performing ensemble methods were developed and used in a wide area of applications, such as stacked generalization [20], mixture of experts [10] and negative correlation learning [8] among others.

In literature, it has been shown that there are two conditions for an ensembles to perform better than a single predictor. These are that the base predictors should be diverse (their error correlation is reduced) and that they have a reasonable level of performance [18]. In ensemble learning diversity has been acknowledged as an important characteristic [6]. An ensemble with diverse models can have better performance due to the complementary behaviour of its components [21], however, as shown in [17] the diversity measure used has to be chosen carefully so it works with the used combiner.

The work presented in this paper builds on our broader investigations of multilevel structures of classifiers and predictors [11,14,16,18] and directly follows from our previous work presented in [1]. It discusses diversity as a characteristic of an MCMLPS, and investigates its effect on the accuracy of prediction.

The organization of this paper is: in Sect. 2 a new type of ensemble system is introduced. Section 3 explores the methodology and the design cycle of the proposed locally trained MCMLPS. Section 4 discusses the experimental work in the paper and the obtained results. It compares the testing accuracy of the system with four benchmark algorithms and studies the relation between the overall accuracy of the ensemble and the amount of disagreements among the base predictors. Section 5 explores a number of variations in the parameters of the proposed systems. Finally Sect. 6 draws the main conclusions in the paper.

2 Multi-Component, Multi-Layer Predictive Systems

The MCMLPS used in this study was introduced in [1] and it is shown in Fig. 1; where $w_{11},...,w_{nk}$, are the weights of the first layer, n represent the number of the base ensembles and k represent the number of the models inside the base ensembles. Furthermore, $w_1,...,w_n$ are the weights of the second layer for the n base ensembles. $M_1,...,M_k$ are the base predictors of the first layer ensembles, $g_1,...,g_n$ are the ensembles created from combining the base predictors, $h(x)$ is the second layer combiner and $\hat{Y}$ is the final prediction of the system. Let X be the data set containing the training objects, C represent the number of classes, θ_c represent the actual class and M_k^n represent the output prediction of the model (shown in the first layer of the ensemble in Fig. 1), where $M_k^n = 1$ for class θ_c and 0 otherwise and $c = 1,..,C$. The outputs of the base predictors M_k^n and the ensemble g_n are given as c-dimensional binary vectors where $[M_1^j,..,M_k^j]^T \in \{0,1\}^c$ and $[g_1,..,g_j]^T \in [0,1]^c$, j=1,...,n respectively. Equations 1 and 2 show the mathematical representation for the ensembles generated from the first layer:

$$g_j(x) = \Sigma_{i=1}^{k} w_{1j} M_i^{(j)}(x) \tag{1}$$

and let

$$d_{j,c}(x) = \begin{cases} 1 & if\ g_j(x) = \theta_c, \\ 0 & otherwise. \end{cases} \tag{2}$$

Then the second layer ensemble is as:

$$h(x) = \Sigma_{j=1}^{m} w_{2j} d_{j,c} \tag{3}$$

and the final prediction of the system is:

$$\hat{Y} = \arg\max_{c} h(x). \tag{4}$$

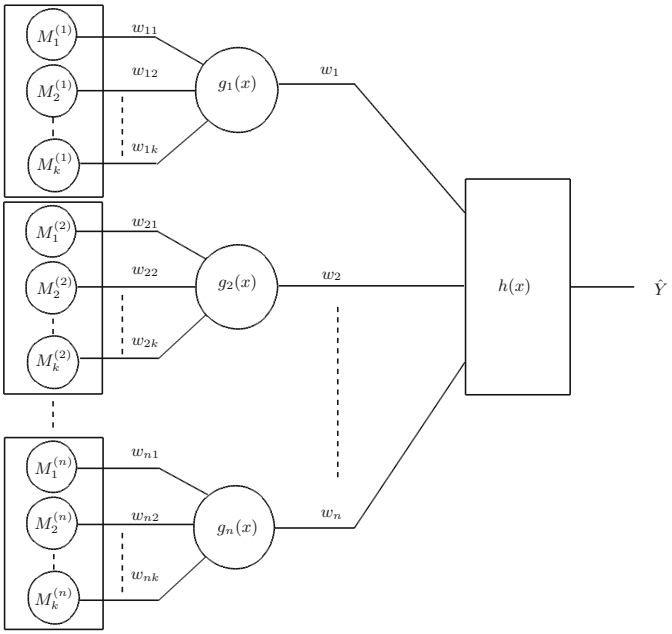

Fig. 1. The multi-component multi-layer predictive system.

3 Designing MCMLPS: Methodology

Despite the similarities between the MCMLPS presented in this paper with that presented in [1], there is a key difference between these systems. The approach introduced in [1] is an unsupervised learning approach, where the base predictors are trained on disjoint sets of the data for which only subsets of the features are selected. Meanwhile, the MCMLPS presented in this paper is a supervised learning approach in which the base predictors are trained on subsets of the features for all of the training data. Moreover, it uses a different similarity metric. The methodology used to built the MCMLPS encompasses the following phases: (a) data preparation and partitioning, (b) model generation and combination.

In order to validate and examine the generalization ability of the proposed architecture, the Density Preserving Sampling (DPS) [5] is used to partition the

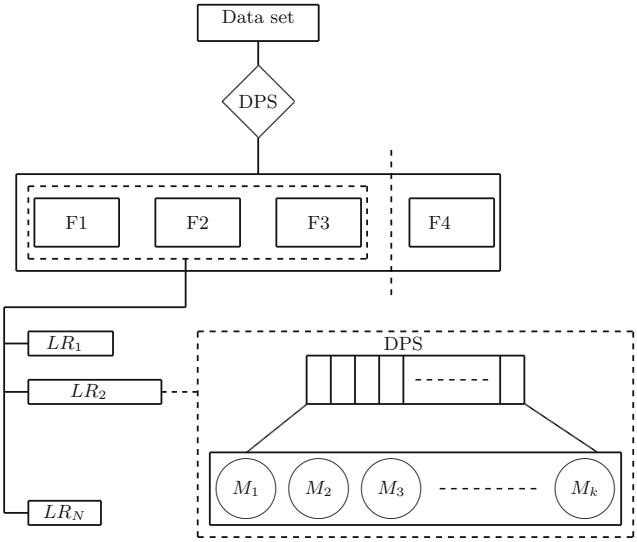

Fig. 2. Data preparation and model generation.

data. DPS divides the data into subsets that are representative of the whole data set [5]. In this work DPS is used to split the data into training and testing sets. The training data is assigned according to its features similarity to a set of LRs. The similarity is determined using mutual information based approach (discussed in Subsect. 3.1). Then DPS is used again to split the LRs data into K folds, where K models are trained on the data of the generated folds. The general design phases for the MCMLPS are discussed below:

– Data preparation and partitioning:
The data goes through three partitioning stages, first the whole data is split into training and testing sets, then the training set is allocated to the LRs and finally within the LRs the data is split into K subsets which are used to train the local models. Figure 2 shows the preparation and partitioning of the data, where, $F_1, ...F_4$ are the folds generated from the first DPS split, $LR_1, ...LR_N$ are the LRs and $M_1, ...M_k$ are the local models within the regions trained using data from the second DPS split.

The points given below summarise the procedure used in this phase:

- Apply DPS to split the data into 4 representative folds.
- Use 3 out of 4 folds for training and the last fold for testing.
- Find the similarity matrix for the training data using the mutual information of the features.
- Choose N rows from the similarity matrix to be the seeds for the LRs.
- Add the training data to the LRs according to the similarity of data features to the LRs seeds.

- Apply k fold DPS to the LRs data.

– Model generation, testing and combining:
Once the data is assigned to the relevant LRs, the second DPS is applied to generate the K folds within the LRs and K models are trained on the LR folds. Furthermore, for all new instances N weights values are computed with respect to the N LRs. This phase can be summarized as follow:

- Train a predictive model on each of the K LRs folds.
- Compute the weights of the LRs votes using the similarity between the LRs seeds and the testing data.

In the first layer, N ensembles are generated from combining the models of the N LRs. While, in the second layer a single ensemble that combines the first layer N ensembles is generated. The combining method used is a weighted majority vote with the similarity of the LRs features used as the weights in both layers. The procedure is repeated for all four folds $F1, ...F4$, so that each time a different fold is used for testing.

3.1 Conditional Mutual Information Based LRs

This approach aims to split the feature space into a number of subsets based on their Conditional Mutual Information (CMI). The features with the highest CMI values are chosen to be the seeds for the LRs. The CMI is measured using the following equation [4]:

$$J_{cmi}(X_k) = I(X_k;Y) - I(X_k;S) + I(X_k,S|Y) \tag{5}$$

where X_k is a single feature, Y is the output and S are the remaining features (all the features apart from X_k). $I(X_k;Y)$ is the mutual information between the feature X_k and the class Y, $I(X_k;S)$ is the redundancy of feature X_k with respect to the remaining features and $I(X_k,S|Y)$ is the conditional redundancy (the class dependency of X_k with the existing feature set S). According to [4] the equation given above shows that including correlated features can be useful, if the correlation of the features with the class is higher than their inner correlation. The benefits of including correlated features have been explored before by [9], where it has been observed that "correlation does not imply redundancy" .

Once the CMI values of the features are computed using Eq. 5, the highest N features are selected to be the seeds for the LRs. In order to add new features to the LRs, the similarity of the features to the LRs seeds need to be calculated. Equation 6 is used to determine the similarity between the features and the LRs seeds.

$$J_{cmi+}(X_k) = I(X_k;Y) + I(X_k;J_{cmi}(X_k)) + I(X_k,J_{cmi}(X_k)|Y) \tag{6}$$

In this equation the pairwise mutual information of the features with the LR seeds is calculated and the features that have the highest CMI with respect to the seeds are added to the LRs. By adding rather than subtracting the redundancy term $I(X_k;J_{cmi}(X_k))$ this approach aims to group together similar features in

the LRs. Each LR is assigned with a subset of the features, where all the features are ranked according to their mutual information with the seed of the LR and only the highest ranking features are assigned to the LR. The ratio of the features assigned to the LRs is α, where $1 > \alpha > 0$.

In order to use this approach to build an MCMLPS, initially the data is split using the method presented in Fig. 2. DPS is also used to split the data into training and testing. Then the following steps are taken to split the training data into the N LRs:

1. Calculate the CMI among the training data features using Eq. 5.
2. Choose the highest scoring N features to be the seeds of the LRs.
3. For the remaining features, use Eq. 6 to rank the features according to their similarity to the LRs seeds.
4. Based on the features mutual information with the seeds, assign α of the total number of features to the LRs.

In both layers weighted majority vote is used to combine the respective predictions, where the mutual information of the LRs features is used as the weighing vector. The weights of the predictions of the LRs models are calculated using the summation of the mutual information values of the LR features.

4 Results

The MI based MCMLPS introduced in this paper is applied to the data sets shown in Table 1. The data sets used are taken from the UCI machine learning archive [12]. The performance of this system is compared to correlation based MCMLPS [1], Rotation Forest (RF) [15], Bagging [3] and AdaBoost [19]. The settings for these benchmark algorithms is as follows:

- MI based MCMLPS: 6 LR's are used with each having 8 models (48 Decision Trees (DT's) in total) trained on α subset of the features.
- Correlation based MCMLPS: 6 LR's are used with each having 8 models trained on disjoint subsets of the data. The number of features used in the LRs is determined through a separate optimization routine [1].
- RF: the number of classifier are 6 and the number of disjoint features subspaces are 6.
- AdaBoost and Bagging: 48 DT were used as the weak learners for both algorithms.

In order to be able to compare the results obtained from this system with the correlation based MCMLPS, both the number of the LRs and the number of models inside the LRs are set to the same numbers (6 LRs with 8 models inside each one of the LRs). Furthermore the α value (the ratio of the features assigned to the LRs) is set to 30% of the features. The base predictors used are CART DTs and feedforward Neural Networks (NNs). The following subsections discuss

Table 1. Data sets used in the experiments.

Data sets	Features	Examples	Classes
Ionosphere	34	351	2
Pima	8	768	2
Wisconsin Breast Cancer (WBC)	30	569	2
Heart	13	270	2
Sonar	60	208	2
Chess	36	3196	2
German credit card	24	1000	2
Spam base	57	4601	2
Gaussian 8D	8	5000	2
Vehicle	18	846	4
Waveform	40	5000	3

the internal accuracies of the LRs base predictors and compare the overall system performance with the benchmark algorithms. This is followed by a subsection that investigates the level of disagreement among the LRs prediction of the proposed MI based MCMLPS and compare its overall performance with benchmark algorithms.

4.1 Internal Accuracy and Benchmark Comparison

In this section the internal accuracies of the LRs base predictors (CART DTs) are measured and compared across the four DPS folds. An example of the LRs base predictors internal accuracies for the Gaussian 8 dimensional data set is shown in Fig. 3. Figure 3 shows that, there are no single LRs that outperform the other LRs on all of the four folds. In the MI approach even small data sets like the Ionosphere data set, has a lower variation in its internal accuracies compared to the results of the correlation based MCMLPS [1]. A possible explanation for this is that the LRs in this case are trained on a subset of the features for the whole data set rather than being trained on disjoint subsets of the data. The overall testing accuracy of the MI based MCMLPS averaged over the four DPS iterations are shown in Table 2. In addition, the Table shows the test accuracies of the four benchmark algorithms (correlation based MCMLPS, RF, Bagging and AdaBoost algorithms). The results show that, this approach for generating the LRs has generally improved the testing accuracy obtained from the correlation based MCMLPS.

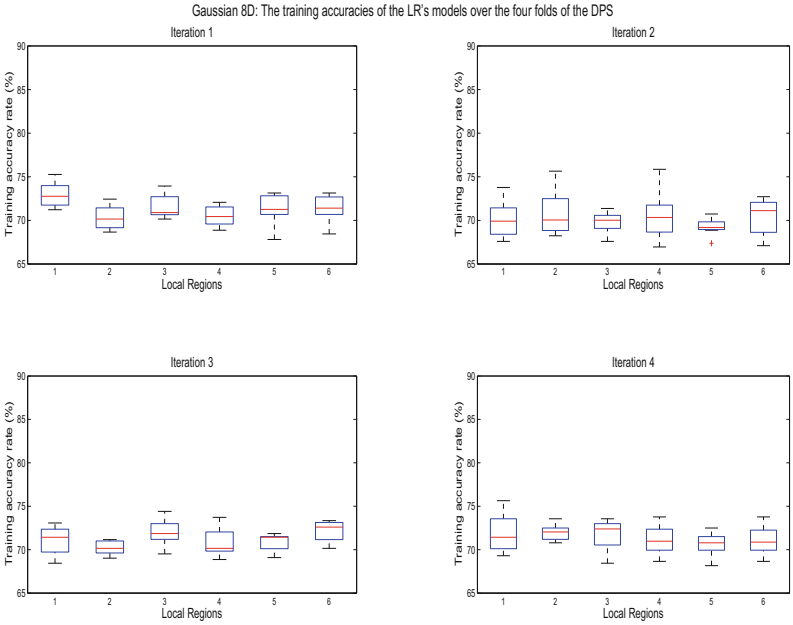

Fig. 3. Training accuracies of the local regions models for the Gaussian 8D data set when CART DTs are used as the base predictors.

Furthermore, it can be seen in Table 2 that Bagging has the highest test accuracy on all the data sets except for the waveform data set, where the RF has the highest accuracy. Nevertheless, our proposed MCMLPS has a comparable accuracy to the Bagging algorithm, with accuracy difference ranges from having the same accuracy for WBC data set to 6.2 for the heart data set. Furthermore, Table 3 shows the test accuracy of the MI based MCMLPS compared to the correlation base MCMLPS and the RF, when the type of the base predictors is changed from CART DTs to feedforward NNs. In the RF algorithm, the testing accuracy increases on every single data set when the feedforward NNs are used as the base predictors. On the other hand, the MI based MCMLPS showed mixed responses, where the accuracy increased for only 4 out of 11 data sets.

4.2 Disagreement Among the Base Predictors

The disagreements among the LRs votes and the final prediction, when CART DTs as well as feedforward NNs are used as the base predictors for the MI based MCMLPs, are shown in Fig. 4. The total disagreement values are found by measuring the disagreement between the final prediction of the system and the prediction of the individual LRs ensembles. In Fig. 4 it can be noticed that, in the proposed architecture, when CART DTs are used as the base predictors there are varied levels of disagreements within the LRs models and even a higher

Table 2. Benchmark comparison: Testing accuracy using CART DTs as the base predictors for both correlation based and MI based MCMLPS.

Data sets	MI based MCMLPS	Correlation based MCMLPS	RF	Bagging	AdaBoost
Gaussian 8D	86.94	88.16	80.70	88.78	87.08
German	74.60	70.00	65.30	77.30	75.90
Ionosphere	92.30	77.19	92.61	93.44	93.16
Spam base	93.81	85.20	85.50	95.37	93.20
Pima	75.78	76.62	73.30	77.60	77.08
WBC	95.61	86.29	91.56	95.61	95.25
Heart	78.89	76.65	77.06	85.18	83.34
Sonar	84.62	63.46	74.04	87.02	83.17
Chess	98.78	93.74	70.46	98.99	94.84
Vehicle	74.35	67.61	61.37	77.07	51.07
Waveform	81.80	65.68	91.46	85.74	80.78

Table 3. Benchmark comparison: Testing accuracy using feedforward NNs as the base predictors for the MI based MCMLPS.

Data sets	MI based MCMLPS	Energy based MCMLPS	RF
Gaussian 8D	84.22	88.45	88.4
German credit cards	77.50	70.00	70.00
Ionosphere	90.03	74.25	93.15
Spambase	90.44	90.55	85.75
Pima indians diabetes	76.04	76.30	76.80
WBC	94.90	91.55	95.61
Heart	82.61	77.02	81.12
Sonar	82.21	62.50	79.81
Chess	94.65	96.75	73.06
Vehicle	79.67	78.50	81.75
Waveform	85.36	85.05	92.65

level of disagreement across the LRs. On the other hand, when feedforward NNs are used as the base predictors, similar models are generated in the individual LRs, yet there is still a high level of disagreement across the LRs. The high level of disagreement of the proposed architecture can be beneficial when applied on noisy data sets.

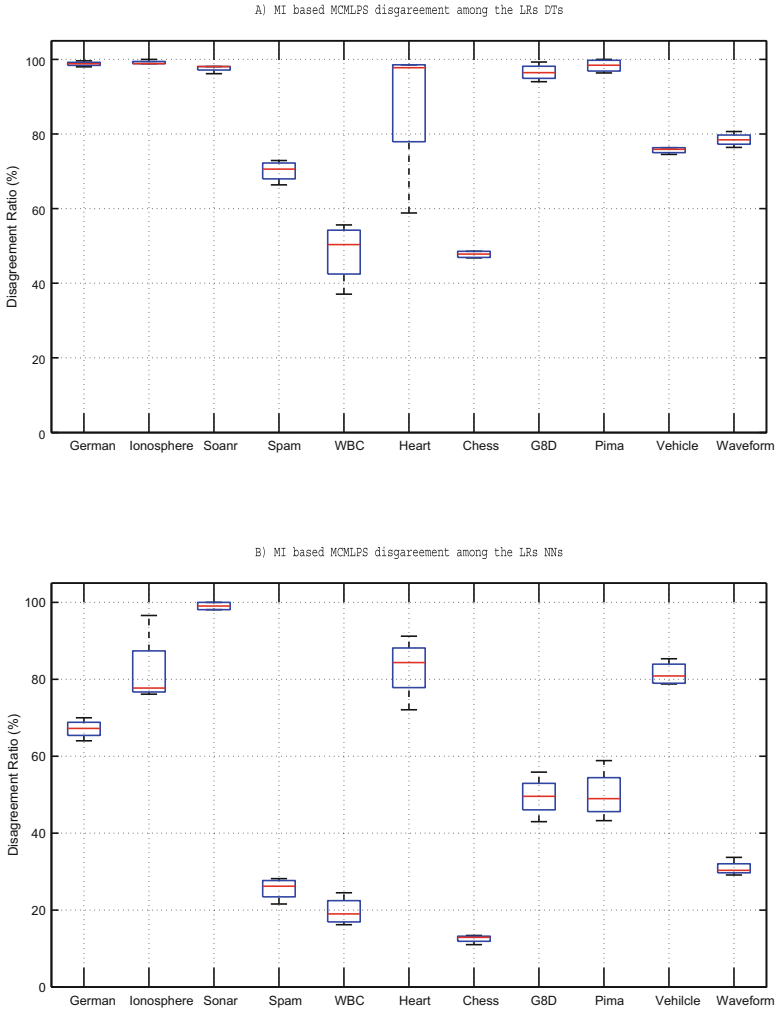

Fig. 4. Disagreements among the LRs of MI based MCMLPS when CART DTs and feedforward NNs are used as the base predictors.

5 Variation of the Conditional Mutual Information

This section investigates the effect of changing three aspects of the proposed MI based architecture. These are: modifying the equation used to find the LR seeds, partitioning the data using Cross Validation (CV) instead of DPS and changing the ratio of features allocated to the LRs. Table 4 compares the testing accuracy for the proposed architecture when the data is sampled using DPS as well as CV and when the conditional redundancy is included or excluded from the CMI equation.

Table 4. Benchmark comparison: Testing accuracy using feedforward NNs as the base predictors for the MI based MCMLPS.

Data sets	DPS with conditional redundancy	DPS ignore conditional redundancy	CV with conditional redundancy
Gaussian 8D	86.94	83.48	84.74
German credit cards	74.60	77.40	76.20
Ionosphere	92.30	91.16	88.28
Spambase	93.81	92.85	92.63
Pima indians diabetes	75.78	72.14	75.52
WBC	95.61	91.20	91.56
Heart	78.89	71.84	77.44
Sonar	84.62	76.92	78.85
Chess	98.78	80.88	95.08
Vehicle	74.35	74.35	63.01
Waveform	81.80	81.70	81.22

5.1 Ignoring the Conditional Redundancy with Respect to the Class

In this case the conditional mutual information term $I(X_k, S|Y)$ is removed from Eq. 5. This transforms the feature selection process to mutual information feature selection proposed by Battiti [2] given in Eq. 7:

$$J_{cmi}(X_k) = I(X_k; Y) - \beta I(X_k; X_j) \tag{7}$$

where β is a configurable parameter for which, according to Battiti [2], the optimal value is often 1. The aim of this section is to compare the case where correlated features are considered as redundant and are removed from the feature selection process with the case where the conditional redundancy between the features is assessed with respect to the class. The results showed that, apart from the German credit card data set, the cases where the conditional redundancy is considered in selecting the features, have higher accuracies than the cases where the conditional redundancy are removed during features selection.

5.2 Using CV Instead of DPS

In this subsection stratified CV is used to partition the data set into training and testing sets and then to partition the LRs data into K folds. Table 4 shows the testing accuracies averaged over the four iterations, and it can be seen that using DPS to split the data produce higher accuracies than that obtained from using stratified CV.

5.3 Changing the Ratio of Features Used in the LRs

In the previous experiments, the ratio of features used in the LRs of the MI based MCMLPS was set to 30%. Using a higher or lower feature ratio have been tested on the data sets used in these experiments. It has been found that lowering this ratio from 30% to 10% decreases the accuracy of the LRs prediction as well as the overall accuracy of the system. Meanwhile, increasing it to 80% result in a slight improving in the prediction accuracy for some of the data sets used in this experiment and it remained unchanged for the rest.

6 Conclusions and Future Work

This paper introduces a local learning based algorithm for MCMLPS. The architecture consists of multiple LRs. Each LR has multiple models trained on subsets of the features. These subsets of features are assigned to the LRs according to the similarity calculated using their conditional mutual information.

Investigating the internal performance of the proposed architecture showed that the overall testing accuracies of the architecture exceeded the average internal accuracies of its LRs models. The amount of variation in the internal accuracy depends mainly on the size and dimensionality of the data. The results showed that both the number of LRs and the number of models developed within the LRs need to be optimised with respect to the data set size and dimensionality.

This paper also explored changing three aspects of the proposed architecture. The first aspect is modifying the equation used to find the LR seeds, where removing the correlation redundancy term from the CMI equation resulted in deterioration of the performance of the proposed architecture. This result support the claim in [4], that including correlated features can be useful if their correlation with the class is higher than their inner correlation. The second aspect is partitioning the data using CV instead of DPS. Changing the sampling technique did have a negative effect on the performance of the proposed architecture, where mainly the accuracy obtained from DPS is higher than that obtained from CV. Finally, increasing the ratio of the features used in the LRs may improve the accuracy of the MCMLPS for certain data sets.

The locality of the proposed architecture and the high level of disagreement among its base predictors can be beneficial in noisy environments. For example, when the noise is applied to only a part of the data, it will not have the same effect on all of the MCMLPS base predictors. The robustness of the proposed architecture to external noise will be investigated in future work.

References

1. Al-Jubouri, B., Gabrys, B.: Local learning for multi-layer, multi-component predictive system. Procedia Comput. Sci. **96**, 723–732 (2016)
2. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Trans. Neural Netw. **5**(4), 537–550 (1994)

3. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
4. Brown, G., Pocock, A., Zhao, M.J., Luján, M.: Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. J. Mach. Learn. Res. **13**, 27–66 (2012)
5. Budka, M., Gabrys, B.: Density-preserving sampling: robust and efficient alternative to cross-validation for error estimation. IEEE Trans. Neural Netw. Learn. Syst. **24**(1), 22–34 (2013)
6. Cunningham, P., Carney, J.: Diversity versus quality in classification ensembles based on feature selection. In: López de Mántaras, R., Plaza, E. (eds.) ECML 2000. LNCS, vol. 1810, pp. 109–116. Springer, Heidelberg (2000). doi:10.1007/3-540-45164-1_12
7. Dasarathy, B.V., Sheela, B.V.: A composite classifier system design: concepts and methodology. Proc. IEEE **67**(5), 708–713 (1979)
8. Eastwood, M., Gabrys, B.: The dynamics of negative correlation learning. J. VLSI Signal Proc. **49**, 251–263 (2007)
9. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature Extraction: Foundations and Applications, vol. 207. Springer, Heidelberg (2008)
10. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**(1), 79–87 (1991)
11. Kadlec, P., Gabrys, B.: Architecture for development of adaptive on-line prediction models. Memet. Comput. **1**(4), 241–269 (2009)
12. Lichman, M.: UCI machine learning repository (2013). http://archive.ics.uci.edu/ml
13. Polikar, R.: Ensemble based systems in decision making. IEEE Circuits Syst. Mag. **6**(3), 21–45 (2006)
14. Riedel, S., Gabrys, B.: Pooling for combination of multi level forecasts. IEEE Trans. Knowl. Data Eng. **12**(21), 1753–1766 (2009)
15. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: a new classifier ensemble method. IEEE Trans. Pattern Anal. Mach. Intell. **28**(10), 1619–1630 (2006)
16. Ruta, D., Gabrys, B., Lemke, C.: A generic multilevel architecture for time series prediction. IEEE Trans. Knowl. Data Eng. **23**(3), 350–359 (2011)
17. Ruta, D., Gabrys, B.: New Measure of Classifier Dependency in Multiple Classifier Systems. In: Roli, F., Kittler, J. (eds.) MCS 2002. LNCS, vol. 2364, pp. 127–136. Springer, Heidelberg (2002). doi:10.1007/3-540-45428-4_13
18. Ruta, D., Gabrys, B.: Classifier selection for majority voting. Inf. Fusion **6**(1), 63–81 (2005)
19. Schapire, R.E.: The strength of weak learnability. Mach. Learn. **5**(2), 197–227 (1990)
20. Wolpert, D.H.: Stacked generalization. Neural Netw. **5**(2), 241–259 (1992)
21. Xue, F., Subbu, R., Bonissone, P.: Locally weighted fusion of multiple predictive models. In: International Joint Conference on Neural Networks, 2006. IJCNN'06, pp. 2137–2143. IEEE (2006)